MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

Mini Project Report
of
Operating Systems LAB

TITLE
"DISK SCHEDULING ALGORITHMS"

SUBMITTED
BY

NAME	REGISTRATION NUMBER	ROLL NUMBER
Abhay Mudgil	200905172	33
Shyam Sundar Bharathi S	200905302	53
Parv Jain	200905316	56
Ajay Rajendra Kumar	200905390	61
Ryan Sojan	200905396	64

INTRODUCTION

- This Project is related to the disk scheduling concept of operating systems.

Disk scheduling is done by operating systems to schedule I/O requests arriving for the disk. Disk scheduling is also known as I/O scheduling. One of the responsibilities of the operating system is to use the hardware efficiently. For the disk drives, meeting this responsibility entails having fast access time and large disk bandwidth.

The seek time is the time for the disk arm to move the heads to the cylinder containing the desired sector. The rotational latency is the additional time for the disk to rotate the desired sector to the disk head. The disk bandwidth is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer. We can improve both the access time and the bandwidth by managing the order in which disk I/O requests are serviced. Whenever a process needs I/O to or from the disk, it issues a system call to the operating system.

If the desired disk drive and controller are available, the request can be serviced immediately. If the drive or controller is busy, any new requests for service will be placed in the queue of pending requests for that drive. For a multiprogramming system with many processes, the disk queue may often have several pending requests. Thus, when one request is completed, the operating system needs to choose which pending request to service next. The operating system makes this choice by implementing any one of the several disk-scheduling algorithms such as FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK, etc.

FCFS

FCFS (First Come-First Serve) is the easiest disk scheduling algorithm among all the scheduling algorithms. In the FCFS disk scheduling algorithm, each input/output request is served in the order in which the requests arrive. In this algorithm, starvation does not occur because FCFS address each request.

The advantages of FCFS disk scheduling algorithm are:

- In FCFS disk scheduling, there is no indefinite delay.
- There is no starvation in FCFS disk scheduling because each request gets a fair chance.

The disadvantages of FCFS disk scheduling algorithm are:

- FCFS scheduling is not offered as the best service.
- In FCFS, scheduling disk time is not optimized.

SSTF

In SSTF (Shortest Seek Time First) disk scheduling, the job which has less seek time will be executed first. So, in SSTF scheduling, we must calculate the seek time first. and after calculating the seek time, each request will be served based on seek time. The request which is close to the disk arm will be first executed. There are some drawbacks to FCFS. To overcome the limitations that arise in the FCFS, SSTF scheduling is implemented.

The advantages of SSTF disk scheduling are:

- In SSTF disk scheduling, the average response time is decreased.
- Increased throughput.

The disadvantages of SSTF disk scheduling are:

- In SSTF, there may be a chance of starvation.
- SSTF is not an optimal algorithm.
- There are chances of overhead in SSTF disk scheduling because, in this algorithm, we must calculate the seek time in advance.
- The speed of this algorithm can be decreased because the direction can be switched frequently.

SCAN

In this algorithm, we move the disk arm in a specific direction (the direction can be moved towards the large value or the smallest value). Each request is addressed that comes in its path, and when it comes to the end of the disk, then the disk arm will move reverse, and all the requests are addressed that are arriving in its path. SCAN disk scheduling algorithm is also called an elevator algorithm because it works like an elevator.

The advantages of SCAN disk scheduling algorithm are:

- In SCAN disk scheduling, there is a low variance of response time.
- In this algorithm, throughput is high.
- Response time is average.
- In SCAN disk scheduling, there is no starvation.

The disadvantages of SCAN disk scheduling algorithm are:

- SCAN disk scheduling algorithm takes a long waiting time for the cylinders, just visited by the head.
- In SCAN disk scheduling, we must move the disk head to the end of the disk even when we don’t have any request to service.

C-SCAN

C-SCAN stands for Circular-SCAN. C-SCAN is an enhanced version of SCAN disk scheduling. In the C-SCAN disk scheduling algorithm, the disk head starts to move at one end of the disk and moves towards the other end and services the requests that come in its path and reach another end. After doing this, the direction of the head is reversed. The head reaches the first end without satisfying any request and then it goes back and services the requests which are remaining.

The advantages of the C-SCAN disk scheduling algorithm are:

- C-SCAN offers better uniform waiting time.
- It offers a better response time.

The disadvantages of the C-SCAN disk scheduling algorithm are:

- In C-SCAN disk scheduling, there are more seek movements as compared to SCAN disk scheduling.
- In C-SCAN disk scheduling, we must move the disk head to the end of the disk even when we don't have any request to service.

LOOK

LOOK scheduling is an enhanced version of SCAN disk scheduling. LOOK disk scheduling is the same as SCAN disk scheduling, but in this scheduling, instead of going till the last track, we go till the last request and then change the direction.

The advantages of LOOK disk scheduling are:

- In LOOK disk scheduling, there is no starvation.
- LOOK disk scheduling offers low variance in waiting time and response time.
- LOOK disk scheduling offers better performance as compared to SCAN disk scheduling as there is no requirement for the disk head to move till the end of the disk when we do not have any request to be serviced.

The disadvantages of LOOK disk scheduling are:

- In LOOK disk scheduling, there is more overhead in finding the end request.
- LOOK disk scheduling is not used in case of more load.

C-LOOK

C-LOOK means Circular-LOOK. It takes the advantage of both the disk scheduling C-SCAN and LOOK disk scheduling. In C-LOOK scheduling, the disk arm moves and services each request till the head reaches its highest request, and after that, the disk arm jumps to the lowest cylinder without servicing any request, and the disk arm moves further and services those requests which are remaining.

The advantages of C-LOOK disk scheduling are:

- There is no starvation in C-LOOK disk scheduling.
- The performance of the C-LOOK scheduling is better than Look disk scheduling.
- C-LOOK disk scheduling offers low variance in waiting time and response time.

The disadvantages of C-LOOK disk scheduling are:

- In C-LOOK disk scheduling there may be more overhead in determining the end request.
- There is more overhead in calculations.

OUTCOME

```
                Disk Scheduling Policies Demo

Do you want to input values or should we do it for you ? [1/0] : 1

Enter length of the request queue : 7

Enter the request queue :
Enter element 1: 82

Enter element 2: 170

Enter element 3: 43

Enter element 4: 140

Enter element 5: 24

Enter element 6: 16

Enter element 7: 190

Enter the current head location : 50
```

FCFS: -

```
Policies you can try out :
1. FCFS
2. SSTF
3. SCAN
4. C-SCAN
5. LOOK
6. C-LOOK
7. EXIT

What would you like [1-7] : 1

                +--------------------------------------+
                |Output for FCFS Disk Scheduling Policy|
                +--------------------------------------+

        Head currently at: 50

     Iteration   Current Head    Disk Movement   Total Disk Movement
-------------------------------------------------------------------------
        1               82              32              32
        2               170             88              120
        3               43              127             247
        4               140             97              344
        5               24              116             460
        6               16              8               468
        7               190             174             642
-------------------------------------------------------------------------

        Total Head Movement: 642

        Order of Processing: 82  -> 170  -> 43  -> 140  -> 24  -> 16  -> 190

        Average Seek Time : 642 / 7 = 91.7143
```

SSTF: -

```
Policies you can try out :
1. FCFS
2. SSTF
3. SCAN
4. C-SCAN
5. LOOK
6. C-LOOK
7. EXIT

What would you like [1-7] : 2

                +--------------------------------------+
                |Output for SSTF Disk Scheduling Policy|
                +--------------------------------------+

        Head currently at: 50

     Iteration   Current Head    Disk Movement   Total Disk Movement
-------------------------------------------------------------------------
        1               43              7               7
        2               24              19              26
        3               16              8               34
        4               82              66              100
        5               140             58              158
        6               170             30              188
        7               190             20              208
-------------------------------------------------------------------------

        Total Head Movement: 208

        Order of Processing: 43  -> 24  -> 16  -> 82  -> 140  -> 170  -> 190

        Average Seek Time : 208 / 7 = 29.7143
```

SCAN: -

```
What would you like [1-7] : 3
Enter the Page Limit: 199
[ 16 24 43 ][ 82 140 170 190 ]
Traverse to the inner track or the outer track?
[0] Inner Track
[1] Outer Track
Your Choice? 0

                +--------------------------------------+
                |Output for SCAN Disk Scheduling Policy|
                +--------------------------------------+

        Head currently at: 50

     Iteration   Current Head    Disk Movement   Total Disk Movement
-------------------------------------------------------------------------
        1               43              7               7
        2               24              19              26
        3               16              8               34
        4               0               16              50
        5               82              82              132
        6               140             58              190
        7               170             30              220
        8               190             20              240
-------------------------------------------------------------------------

        Total Head Movement: 240

        Order of Processing: 43 -> 24 -> 16 -> 0 -> 82 -> 140 -> 170 -> 190

        Average Seek Time : 240 / 7 = 34.2857
```

```
What would you like [1-7] : 3
Enter the Page Limit: 199
[ 16 24 43 ][ 82 140 170 190 ]
Traverse to the inner track or the outer track?
[0] Inner Track
[1] Outer Track
Your Choice? 1

               +--------------------------------------+
               |Output for SCAN Disk Scheduling Policy|
               +--------------------------------------+

       Head currently at: 50

    Iteration   Current Head    Disk Movement   Total Disk Movement
-------------------------------------------------------------------------
       1              82             32             32
       2              140            58             90
       3              170            30             120
       4              190            20             140
       5              199            9              149
       6              43             156            305
       7              24             19             324
       8              16             8              332
-------------------------------------------------------------------------

       Total Head Movement: 332

       Order of Processing: 82 -> 140 -> 170 -> 190 -> 199 -> 43 -> 24 -> 16

       Average Seek Time : 332 / 7 = 47.4286
```

CSCAN: -

```
What would you like [1-7] : 4
Enter the page limit: 199
[ 16 24 43 ][82 140 170 190 ]
Traverse to the inner track or the outer track?
[0] Inner Track
[1] Outer Track
Your Choice? 0

               +--------------------------------------+
               |Output for C-SCAN Disk Scheduling Policy|
               +--------------------------------------+

       Head currently at: 50

    Iteration   Current Head    Disk Movement   Total Disk Movement
-------------------------------------------------------------------------
       1              43             7              7
       2              24             19             26
       3              16             8              34
       4              0              16             50
       5              199            199            249
       6              190            9              258
       7              170            20             278
       8              140            30             308
       9              82             58             366
-------------------------------------------------------------------------

       Total Head Movement: 366

       Order of Processing: 43 -> 24 -> 16 -> 0 -> 199 -> 190 -> 170 -> 140 -> 82

       Average Seek Time : 366 / 7 = 52.2857
```

```
What would you like [1-7] : 4
Enter the page limit: 199
[ 16 24 43 ][82 140 170 190 ]
Traverse to the inner track or the outer track?
[0] Inner Track
[1] Outer Track
Your Choice? 1

               +--------------------------------------+
               |Output for C-SCAN Disk Scheduling Policy|
               +--------------------------------------+

       Head currently at: 50

     Iteration   Current Head    Disk Movement   Total Disk Movement
-------------------------------------------------------------------------
       1               82              32              32
       2               140             58              90
       3               170             30              120
       4               190             20              140
       5               199             9               149
       6               0               199             348
       7               16              16              364
       8               24              8               372
       9               43              19              391
-------------------------------------------------------------------------

       Total Head Movement: 391

       Order of Processing: 82 -> 140 -> 170 -> 190 -> 199 -> 0 -> 16 -> 24 -> 43

       Average Seek Time : 391 / 7 = 55.8571
```

LOOK: -

```
What would you like [1-7] : 5
[ 16 24 43 ] [ 82 140 170 190 ]
Traverse to the inner track or the outer track?
 [0] Inner Track
 [1] Outer Track
 Your Choice? 0

               +--------------------------------------+
               |Output for LOOK Disk Scheduling Policy|
               +--------------------------------------+

       Head currently at: 50

    Iteration   Current Head    Disk Movement   Total Disk Movement
-------------------------------------------------------------------------
       1              43             7              7
       2              24             19             26
       3              16             8              34
       4              82             66             100
       5              140            58             158
       6              170            30             188
       7              190            20             208
-------------------------------------------------------------------------

       Total Head Movement: 208

       Order of Processing: 43 -> 24 -> 16 -> 82 -> 140 -> 170 -> 190

       Average Seek Time : 208 / 7 = 29.7143
```

```
What would you like [1-7] : 5
[ 16 24 43 ] [ 82 140 170 190 ]
Traverse to the inner track or the outer track?
 [0] Inner Track
 [1] Outer Track
 Your Choice? 1

               +-------------------------------------+
               |Output for LOOK Disk Scheduling Policy|
               +-------------------------------------+

       Head currently at: 50

     Iteration   Current Head    Disk Movement   Total Disk Movement
-------------------------------------------------------------------------
       1               82              32              32
       2               140             58              90
       3               170             30              120
       4               190             20              140
       5               43              147             287
       6               24              19              306
       7               16              8               314
-------------------------------------------------------------------------

       Total Head Movement: 314

       Order of Processing: 82 -> 140 -> 170 -> 190 -> 43 -> 24 -> 16

       Average Seek Time : 314 / 7 = 44.8571
```

CLOOK: -

```
What would you like [1-7] : 6
[ 16 24 43 ] [ 82 140 170 190 ]
Traverse to the inner track or the outer track?
 [0] Inner Track
 [1] Outer Track
 Your Choice? 0

               +---------------------------------------+
               |Output for C-LOOK Disk Scheduling Policy|
               +---------------------------------------+

       Head currently at: 50

    Iteration   Current Head    Disk Movement   Total Disk Movement
-------------------------------------------------------------------------
       1              43             7              7
       2              24             19             26
       3              16             8              34
       4              190            174            208
       5              170            20             228
       6              140            30             258
       7              82             58             316
-------------------------------------------------------------------------

       Total Head Movement: 316

       Order of Processing: 43 -> 24 -> 16 -> 190 -> 170 -> 140 -> 82

       Average Seek Time : 316 / 7 = 45.1429
```

```
What would you like [1-7] : 6
[ 16 24 43 ] [ 82 140 170 190 ]
Traverse to the inner track or the outer track?
 [0] Inner Track
 [1] Outer Track
 Your Choice? 1

               +---------------------------------------+
               |Output for C-LOOK Disk Scheduling Policy|
               +---------------------------------------+

       Head currently at: 50

    Iteration   Current Head    Disk Movement   Total Disk Movement
-------------------------------------------------------------------------
       1              82             32             32
       2              140            58             90
       3              170            30             120
       4              190            20             140
       5              16             174            314
       6              24             8              322
       7              43             19             341
-------------------------------------------------------------------------

       Total Head Movement: 341

       Order of Processing: 82 -> 140 -> 170 -> 190 -> 16 -> 24 -> 43

       Average Seek Time : 341 / 7 = 48.7143
```

CONCLUSION

SSTF is common and has a natural appeal because it increases performance over FCFS. SCAN and C-SCAN perform better for systems that place a heavy load on the disk because they are less likely to cause a starvation problem. For any list of requests, we can define an optimal order of retrieval, but the computation needed to find an optimal schedule may not justify the savings over SSTF or SCAN.

With any scheduling algorithm, however, performance depends heavily on the number and types of requests. For instance, suppose that the queue usually has just one outstanding request. Then, all scheduling algorithms behave the same, because they have only one choice of where to move the disk head: they all behave like FCFS scheduling.

Requests for disk service can be greatly influenced by the file-allocation method. A program reading a contiguously allocated file will generate several requests that are close together on the disk, resulting in limited head movement. A linked or indexed file, in contrast, may include blocks that are widely scattered on the disk, resulting in greater head movement.

The location of directories and index blocks is also important. Since every file must be opened to be used, and opening a file requires searching the directory structure, the directories will be accessed frequently. Suppose that a directory entry is on the first cylinder and a file's data are on the final cylinder. In this case, the disk head must move the entire width of the disk. If the directory entry were on the middle cylinder, the head would have to move only one-half the width. Caching the directories and index blocks in the main memory can also help to reduce disk-arm movement, particularly for read requests. Because of these complexities, the disk-scheduling algorithm should be written as a separate module of the operating system, so that it can be replaced with a different algorithm if necessary. Either SSTF or LOOK is a reasonable choice for the default algorithm.

REFERENCES

- https://www.tutorialandexample.com/scan-disk-scheduling-algorithm